



Agregando Spring Security

Transcripción

[00:00] Bien, llegó la hora más interesante, vamos a entrar a la parte técnica en este momento. Como les dije, lo primero que vamos a hacer ahora es instalar mi módulo de Spring Security.

[00:10] Al igual que yo lo hice por ejemplo con mis otras dependencias como devtools, Spring starter web, ahora también necesito instalar mi módulo de Spring Security aquí en mi API. ¿Recuerdan el truco que les mostré en el curso anterior para instalar dependencias aquí? Bueno, por si no lo vieron lo vamos a ver en este momento.

[00:32] Para eso es lo que voy a hacer es entrar aquí a Spring initializr, en start.spring.io, y de aquí me interesa, aquí voy a seleccionar Maven y voy a agregar una dependencia. Doy clic aquí, tengo mi lista de dependencias aquí en Spring Initializr y yo voy a buscar por security.

[00:54] Observen allí, Spring Security es la primera opción. Tengo otras opciones también como el OAuth2 Client, en caso yo desee autenticar usando un third party por ejemplo, como puede ser Microsoft, GitHub, Google, no se preocupen, eso no lo vamos ahora.

[01:12] Okta, Spring LDAP, eso es para otro tipo de autenticación con un servidor. No se preocupen, no se marean vamos de frente a Spring Security. Entramos aquí. ¿Y en qué consiste el truco? En lugar de generar el proyecto lo que yo voy a hacer es darle a Explore para ver cómo es que me generaría el

proyecto aquí en la web, y dentro de aquí del pom.xml, yo voy a buscar por el módulo Spring Security que está aquí.

[01:44] Spring-boot-starter-security. Entonces voy a copiar esto, vuelvo a mi proyecto y recuerden siempre: dentro del arreglo de dependencias siempre guíense por el IDE o como los ayude, esta línea me dice hasta dónde llega mi arreglo de dependencias, aquí al final voy a agregar mi dependencia de Spring Security y lo que falta hacer ahora ¿qué es?

[02:09] Refrescar mis dependencias, porque yo aún no las he instalado. Lo que voy a hacer es ejecutar este botón de Maven para que comience a resolver las dependencias y voy a revisar aquí en mi lista si ya tengo la dependencia de Security. Y observen aquí al final, springframework.boot-spring-boot-starter-security.

[02:33] Entonces ya tengo todo por ahora, ya tengo mi Spring Security, ya tengo la dependencia instalada y ya la tengo descargada de Maven Repository. Voy a cerrar esto de aquí y lo que voy a hacer ahora es iniciar mi servidor para ver qué novedades trae. Estaba cargando hasta hace un momento, vamos a ver si sale algo diferente ahora.

[02:53] Esperamos un momento que cargue, vemos que está cargando. Y hay algo muy interesante aquí que quiero que observen, miren aquí. Acá dice usando contraseña generada, contraseña generada y me da un carácter alfanumérico un poco grande, me dice “esta clave es generada solo para uso de desarrollo”.

[03:19] Mi configuración de seguridad debe ser autorizada antes de correr la aplicación en producción. ¿Qué significa esto? No se preocupen, lo vamos a ver más adelante, pero ya vemos que solamente con instalar el módulo de Spring Security ya nos está generando una contraseña, aún no sabemos para qué, y ya nos está diciendo que bueno, parece que nuestros recursos si me está

generando una contraseña de alguna forma mis recursos ya deben estar protegidos.

[03:44] Voy a venir a Insomnia y voy a volver a pedir mi lista de médicos que está funcionando hasta hace un momento, voy a entrar y me da un 401 unauthorized. El código unauthorized significa que mi usuario o yo no tengo acceso a este recurso, entonces si se dan cuenta aquí, por ejemplo también para obtener datos de médico, ese no existía, pero vamos a poder uno que existe creo que número 9. Igual. Unauthorized.

[04:14] Entonces, de por sí solamente instalando la dependencia de Spring Security ya me dice que automáticamente por default todos mis requests están bloqueados. Esto en el browser lo vamos a poder ver un poco mejor porque hay una cosa que les quiero mostrar. Voy a copiar esto. Voy a venir a mi browser y voy a copiar esta dirección.

[04:40] Voy a entrar y observen, miren cómo entrando a la dirección automáticamente me manda a /login. Yo no he creado eso, yo no tengo ningún controller con /login. Observen en mi carpeta de controllers, yo no tengo ninguna con login. Solo tengo Médico y HelloControllers que lo creamos en el curso anterior y me da una página en HTML y pide por favor que accese.

[05:09] Entonces aquí entra algo curioso. ¿Por qué? Este tipo de autenticación es autenticación por defecto para aplicaciones web. Aplicaciones web que por defecto me van a generar una sesión del receptor, una aplicación ¿cómo se llamaban? Stateful, exacto, autenticación stateful.

[05:29] Pero ya hemos quedado que nosotros vamos a hacer stateless, vamos a trabajar con la autenticación a nivel del API. Por lo tanto este login a mí no me sirve para nada, es autogenerado por Springs, sí, es completamente autogenerado por Spring Boot, yo no he hecho nada.

[05:49] Él me da por defecto esto porque automáticamente todos mis recursos dentro de mi API ya están protegidos. Ahora ustedes preguntarán: “Okay, ¿y

cómo me logueo entonces? ¿Cuál es la clave por defecto?” Ahí es donde entra esta clave de aquí. Vamos a copiar esta clave, "Ctrl + C".

[06:05] Y el nombre de usuario por defecto aquí en Spring Security se llama user. Entonces user, esta es su contraseña, entramos, le voy a decir que no quiero guardar la página y me dice que okay, no me muestra una página de login, no me está mostrando lo que yo le pedí porque no está digamos completando con el request de médicos.

[06:37] Entonces lo que voy a hacer es venir aquí, copiar y vemos que tengo todo mi arreglo de médicos. ¿Por qué me salió la página anterior de error? Porque yo no tengo ninguna página configurada a nivel de Spring que me diga: “okay, Diego después de loguearte, cuando sea exitoso, la siguiente página es esta de aquí”.

[07:04] Por defecto cuando te logueas vía web que muestra un home de algún dashboard, alguna cosa así. Yo no tengo nada de eso configurado, porque solo son mis recursos de Spring Boot auto generados por Spring Security pero ya vemos que ahora sí tengo acceso porque ya estoy autenticado.

[07:21] Ya tengo acceso a mis recursos, a mi arreglo de JSON porque simplemente ya me autenticué usando la clave por defecto, generada aquí. Nuevamente este no es el tipo de autenticación que nosotros queremos.

[07:35] Lo que nosotros queremos es hacer una aplicación stateless que funcione a nivel de API en el cual obviamente yo no voy a necesitar una página de login porque mis clientes, ya sea aplicación web, mobile o incluso mi cliente Insomnia deberían poder generar la autenticación, mi token, loguearme vía API, generar mi token de autorización para yo conseguir hacer los requests enviando ese token en cada request que yo haga.

[08:06] ¿Suena confuso? No se preocupen, lo vamos a ver poco a poco, paso a paso, en los siguientes videos de esta clase. Nos vemos.

